Remarks

The applicants have carefully reviewed the Office action dated April 3, 2007, and the art applied therein to the claims. Claims 1-6, 8-13, and 15-20 remain in this application, and independent claims 1, 8, and 15 are amended. No new matter has been added. In view of the foregoing amendments and the following remarks, reconsideration and allowance of the application are respectfully requested.

The Rejections Under 35 U.S.C. § 112

In the Office action, claims 7, 14, and 18 were rejected as allegedly indefinite. The applicants respectfully submit that the term "host-protected architecture," as recited in claims 7, 14, and 18, would have been understood by persons having ordinary skill in the art at the time the instant application was filed with the United States Patent and Trademark Office (i.e., December 12, 2003). To illustrate, the applicants file herewith a Supplemental Information Disclosure Statement (IDS) that includes references describing a host-protected architecture, such as a host-protected area (HPA), prior to December 12, 2003.

In particular, the IDS filed herewith includes a standard entitled "Information Technology – AT Attachment with Packet Interface Extension (ATA/ATAPI-4)," published by the American National Standards Institute (ANSI) and dated August 19, 1998 (hereinafter "the standard"). Applicants respectfully invite the examiner to refer to section 6.12 (page 35), which describes an HPA and example applications thereof.

Additionally, the IDS includes a journal article entitled "Hidden Disk Areas: HPA and DCO" by the International Journal of Digital Evidence (hereinafter "the journal") that, among other things, acknowledges that the HPA was introduced in an ATA-4 standard circa 2001 (see the journal, page 2). Furthermore, the IDS includes at least one patent reference having a filing date prior to December 12, 2003 that describes an HPA.

Accordingly, the applicants respectfully submit that persons having ordinary skill in the art would understand the term "host-protected architecture" as recited in claims 7, 14, and 18.

Thus claims 7, 14, and 18 are not indefinite in view of 35 U.S.C. §112, second paragraph, and any rejections based thereon must be withdrawn. Favorable reconsideration is requested.

The Rejections Under 35 U.S.C. § 103(a)

In the Office action, claims 1-20 were rejected as being unpatentable over "Extensible Firmware Interface Specification," Intel Corporation, Version 1.10 (hereinafter "EFI") "in view of obviousness." The applicants respectfully traverse this rejection and request the examiner to clarify what is meant by the claims being unpatentable "in view of obviousness."

Independent claim 1 recites, *inter alia*, booting an operating system to a runtime environment and receiving a pre-boot code update. To establish a *prima facie* case of obviousness, the prior art must teach or suggest each of the claim elements and must additionally provide a suggestion of, or an incentive for, the claimed combination of elements. See *In re Oetiker*, 24 USPQ. 2d 1443, 1446 (Fed. Cir. 1992); *Ex parte Clapp*, 227 USPQ. 972, 973 (Bd. Pat. App. 1985); *In re Royka*, 490 F.2d 981 (CCPA 1974) and M.P.E.P. § 2143. As explained below, the applicants respectfully submit that EFI necessarily fails to include booting an operating system to a runtime environment and receiving a pre-boot code update. Furthermore, the applicants identify additional limitations of independent claim 1 not described by EFI, thereby rendering EFI as a reference that fails to render the claimed subject matter obvious, either alone or in combination with "obviousness."

Unlike the subject matter recited in amended claim 1, EFI describes services (e.g., function calls) related to a boot environment to merely load contents that are already stored in a file path. EFI, pages 1-3, 5-78, 5-79. In particular, EFI identifies the LoadImage() function as a service present in an EFI-compliant system before an operating system is booted. Id. As such, EFI fails to describe an update, receiving a pre-boot code update, much less booting an operating system to a runtime environment prior to receiving the pre-boot code update, as recited in claim 1.

Even if EFI were construed to employ a pre-boot code update, an assertion to which the applicants do not agree, independent claim 1 also recites, *inter alia*, storing the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space in the first non-volatile memory. While EFI describes a LoadImage() function to operate with a system before an operating system is booted and load an image into memory, EFI fails to pay any regard to whether such image will fit within an allocated space. EFI, page 5-79. On the other hand, EFI describes status codes to indicate, for example, whether the function attempt was successful or whether an image was not successfully loaded due to insufficient resources. Id. As a result, unlike storing the pre-boot code update if the pre-boot code update fits, as recited in claim 1, the LoadImage() function of EFI appears to be employed regardless of whether or not the image will fit within an allocated space.

To that end, EFI fails to describe booting an operating system to a runtime environment, a pre-boot code update, much less receiving a pre-boot code update, and storing the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space in the first non-volatile memory.

Independent claim 1 recites, *inter alia*, setting an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update. While EFI employs a "BootAuthorizationCheckFlag" associated with boot integrity services (BIS) and access protocols for network devices, the flag of EFI is not an indication that an update is to be implemented, much less a pre-boot code update. EFI, pages 15-69, 15-89, 15-90. Instead, the flag is merely an operand that enables or disables a check of a digital signature of a data block against a digital certificate. Id. at 15-90. Unlike setting an indication that an update is to be implemented, much less implemented in response to storing the pre-boot code updates, EFI employs the flag for the purpose of an integrity and authorization check. Id. at 15-69.

As neither EFI nor any other cited reference includes booting an operating system to a runtime environment, receiving a pre-boot code update, storing the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space in the first non-volatile memory, or setting an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update, as receited in claim 1, it follows that neither EFI nor any other cited reference, alone or in combination, can render claim 1 obvious. Dependent claims 2-6 depend from independent claim 1 and are allowable for at least the reasons discussed above. Therefore, the applicants respectfully request allowance of claims 1-6.

Independent claim 8, as amended, is also allowable over the art of record for reasons similar to those set forth above in connection with independent claim 1. Specifically, claim 8 recites, *inter alia*, an article of manufacture comprising a machine-accessible medium having a plurality of machine accessible instructions that, when executed, cause a machine to boot an operating system to a runtime environment, receive a pre-boot code update, store the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space, and set an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update. The applicants submit that the rejection of claim 8, and claims 9-13 dependent thereon, must be withdrawn for at least the reasons set forth above in connection with claim 1.

Independent claim 15, as amended, is also allowable over the art of record for reasons similar to those set forth above in connection with independent claim 1. Specifically, claim 15 recites, *inter alia*, a system comprising a processor programmed to boot an operating system to a runtime environment, receive a pre-boot code update, store the pre-boot code update to the flash memory if the pre-boot code update fits within an allocated space, and set an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update. The applicants submit that the rejection of claim 15, and claims 16, 17, 19, and 20 dependent thereon, must be withdrawn for at least the reasons set forth above in connection with claim 1.

Conclusion

Reconsideration of the application and allowance thereof are respectfully requested. If there is any matter that the examiner would like to discuss, the examiner is invited to contact the undersigned representative at the telephone number set forth below.

Respectfully submitted,

Hanley, Flight & Zimmerman, LLC
150 S. Wacker Drive
Suite 2100
Chicago, Illinois 60606

Dated: July 3, 2007

/Mark G. Hanley/

Mark G. Hanley
Reg. No. 44,736
Attorney for Applicants
312.580.1020